

Amendment to Application Claims as Requested by USPTO 5/30/2006

Claim 1:

We claim a method of creating, encapsulating, systematizing, and using information about constituent parts of a language (realized - i.e. visible - and rule based - i.e. no physical representation other than acting on other constituent parts) called the "language-object" which contains "existence" and "appearance" states and exists as an object in the sense of the term as understood in the field of object oriented programming. The language object contains data about its representation and rules that can be applied to said data and other language objects, and data and rules these are classified either as existence states or appearance states:

1. Existence states:

- a. Describe the environment in which a language object may appear
- b. Define possible environments in terms of a language object's relation to other language objects.

2. Appearance States:

- a. Describe what a language object "does" - that is, what rules it may apply to other objects, what relative meanings it may take, what language objects it may require or act as an immediate super-constituent of. (how a language object operates with and on other language objects)
- b. Contain information regarding the scenarios in which a language object's actions, rules, and meanings occur. That is, combine existence state information with the rules, processes, and semantic information contained in appearance states.

Claim 2:

We claim a generalized rule engine for creating language rules for and using language objects based upon a Monte Carlo Markov Chain process for deriving said language objects and information about them and there associations with other language objects.

1. Tests generated by the Markov Process can be based upon:

- a. proximity of one language object to another on a tree structure (defined by the pre-existing network of constituent relationships in the language objects)
- b. existence of other objects with relation to an object
- c. application of a language object rule (that is, a trigger for a new rule based on the application of another rule)

2. Tests and applications of the rules (that is, whether or not a certain rule should be applied given a set of circumstances) are probabilistically based, in that repeated checks form tests that are reliable, or accurate with a high degree of probability.

3. Creation of tests:

a. Observe an occurrence (an “existence” state) of a language object or character. This existence state is a method of describing a possible reason for the appearance of a specific language object.

b. Test this existence state across many appearances of the language object; in this manner, eliminate rules which have a low degree of success for describing the language system, and make prominent those rules with a high degree of success for describing the system.

c. Re-tests of the rules over time to ensure they are still valid within the newly updated system.

Claim 3:

We claim a process for morpheme derivation of a language system based upon usage of language objects and the generalized rule engine:

1. Use proximity tests to begin to derive prevalence of co-occurrence between the base set of characters in the language system being analyzed.
2. Using these proximity tests, create language objects which correspond to the morphemes of a language. These objects will necessarily contain information about the ability of a morpheme to exist in all of its natural environments (the environments in the corpus being analyzed).

Dependent Claim 1 of Claim 3:

We claim a process for word and word phrase derivation of a language system:

1. Apply the language objects created in the morpheme tests to each other, creating rules about the formation of words. The word system of a language will necessarily include morphological process information about the language - how words are formed, which morphemes are allowed next to each other, and so on. (Determine the allowable groupings of morphemes into words, creating rules about the interactions of morphemes in the formation of words.)
2. Using the word objects, create phrasal objects containing multiple words. These rules will necessarily encompass common multi-word phrases (or multi-word-entities) in a language.
3. Both of processes 1 and 2 will begin to incorporate semantic data about the words and multi-word-entities of a language, due to the creation of occurrence-relationship rules for words and word phrases within larger phrase structures.

Dependent Claim 2 of Claim 3:

We claim a process for syntactic derivation of a language system based upon usage of language objects and the generalized rule engine:

1. Given the multi-word-entities and words, create language objects representing phrasal structures.
2. Due to the nature of the system, phrasal structures which appear to function in very limited circumstances will only be allowed in those circumstances, while more generalized phrasal structures (such as the common 'inflectional phrase' or 'complementizer phrase') will have a much greater prevalence, as they describe significantly more data in the language system.
3. Since none of the phrasal language objects are named, it will often be useful, though not necessary, to allow a human to assist the rule-engine in naming them. This will allow a more natural analysis of the phrasal structure, while not taking for granted human tagging or intervention.

Dependent Claim 3 of Claim 3:

We claim a process for semantic derivation of a language system (and its body of free text which is not manually marked up or tagged by humans) based upon usage of language objects and the generalized rule engine:

1. Given phrasal structures, component structures, words and multi-word-entities (in fact, any language object, including major classes of words can be used), the rule engine will posit the existence of semantic meaning structures. This differs from tests of simple co-occurrence relations for semantic analysis, like in Latent Semantic Analysis; unlike in those scenarios, the rule-engine will have an excellent notion of the sub-structure of sentences and phrasal units, as well as full word-sense disambiguation capabilities based on the environments of the word or multi-word-entity language object with respect to the other language objects in the system.
2. The rule engine will then create a semantic map, within the appearance states of the language objects, that describes the usage of the various language objects within the language system. This map is based upon rules contained within multi-word-entity and word objects (all local language objects that the rule engine has deemed to have an effect), as well as phrasal constituent objects that describe the "application" of these objects to other objects. An example of this is a language object describing the NP structure, which could contain a language object describing the AP structure, which would contain a rule describing how the contained adjective/adverb is applied to the NP super-structure.